Homework 7: Structural Induction, Regexes, CFGs

Due date: Wednesday March 1st at 10 PM
If you work with others (and you should!), remember to follow the collaboration policy outlined in the syllabus.

In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.

We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. You are allowed to have longer explanations, but explanations significantly longer than necessary may receive deductions.

To help with formatting of English proofs, we've published a style guide on the website containing some tips. **Unless otherwise noted in a problem, all proofs must be English proofs.** You should not have numbered steps (e.g., you should not be doing an inference proof.)

Finally, be sure to read the grading guidelines for more information on what we're looking for.

1. Walk the walk, talk the talk, log the log [20 points]

Let S be a subset of $\mathbb{Z} \times \mathbb{Z}$ defined recursively as:

Basis Step: $(2, 4) \in S$
Recursive Step: if $(a, b) \in S$ then $(a, 2b) \in S$ and $(2a, 3b) \in S$

Prove that $\forall (a, b) \in S, 2\log_2(a) \leq 1 + 2\log_2(b)$. Note that we use **log base 2** here, also known as binary logarithm.

Hint: Remember that with structural induction you must show $P(s)$ for every element s that is added by the recursive rule – you will need to show $P()$ holds for two different elements in your inductive step.

Hint: The inequality you're showing in your inductive step isn't always tight (that is it could be that $<$ is true, not just $\leq$), remember to keep your target in mind!

Hint: Depending on your proof, you might want these log identities (you can use these without proving them, but do say where you use them if you do).
Fact 1: $c\log_2(d) = \log_2(d^c)$
Fact 2: $\log_2(c \cdot d) = \log_2(c) + \log_2(d)$

2. Chess [23 points]

You and your friend have gotten bored with regular chess and want to play something new. Your friend proposes a new "rook moving game."

To start, the rook[1] is placed at any square on the main lower-left-to-upper-right diagonal (except for the bottom-left corner). You take turns in playing the game. In their turn, each player moves the rook. The player that moves the rook into the bottom-left corner will win the game. A rook can move any number of squares in a straight line, but for this game it can only move left or down (and only one of those in a given player's move).

A sample game is shown in the figures below:

We'll use a generalization of this game: we're now moving the rook on a 2D Cartesian coordinate system. The rook's position is (x, y), and we have $x \geq 0, y \geq 0$. The player that moves the rook to coordinate (0,0) (which is the bottom-left corner) will win the game. In this definition, the coordinate system extend infinitely (x and y can be arbitrarily large), so we are not limited to a standard chessboard anymore. For example, (20,20) would be a valid starting point of the rook. For reference, in the sample game above, the rook starts at (4,4).

[1] A rook is a standard chess piece. You don't need to understand chess to do this problem.

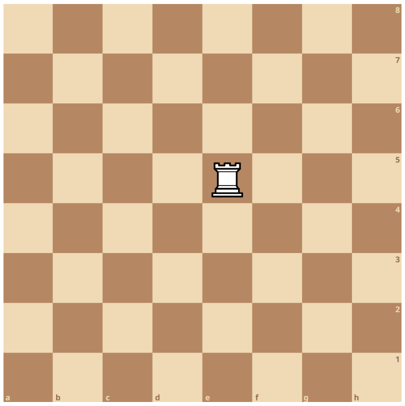

Figure 1: The start of a new game

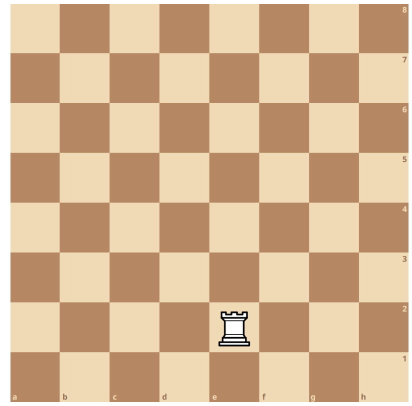

Figure 2: Your friend moves the rook down 3 squares.

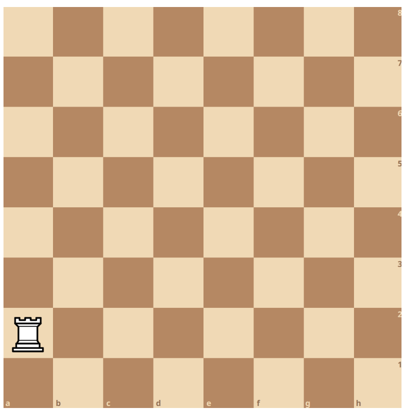

Figure 3: You move the rook left 4 squares.

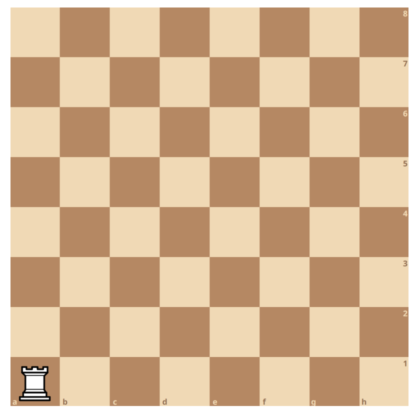

Figure 4: Your friend moves the rook down 1 square. Since the rook is now in the bottom-left square, they win the game.

Note: you don't need any knowledge of chess in order to do this problem. The only thing you need to know is that rook can move any distance on a straight line and in this game it is only allowed to move downward or to the left.

(a) Using induction, prove that in any rook moving game where the rook starts at a coordinate in the form of (a,a) except (0,0), you (the player that goes second) can always win the game.
Be sure to explicitly and clearly define a predicate $P()$! We **strongly** recommend your predicate includes the phrase "It is not my turn" or something similar. [20 points]

(b) Describe your winning strategy (i.e. describe how you should play the game in order to win, assuming that you go second). A strategy would be something like "If my friend moves the rook to location (a, b) then on my turn I will move it to..." [3 points]

3. What doesn't kill you makes you stronger [24 points]

In this problem, we'll use a technique called "strengthened induction." The idea behind the technique is that sometimes the predicate, $P()$, that you want to prove isn't well-suited to induction. $P()$ might not have enough information so that when you use your inductive hypothesis you don't know enough to show $P(k + 1)$. In cases like these, we will sometimes strengthen the predicate.
Consider the following recursively defined functions:

$$f(n) = \begin{cases} 3 & \text{if } n = 1 \\ 4f(n-1) & \text{if } n \in \mathbb{N}, n > 1 \end{cases}$$

$$g(n) = \begin{cases} 2 & \text{if } n = 1 \\ 4g(n-1) + 2 & \text{if } n \in \mathbb{N}, n > 1 \end{cases}$$

Let $P(n)$ be "$g(n) < f(n)$"

(a) Imagine you wanted to prove $P(n)$ for all $n \in \mathbb{N}, n \geq 1$ by induction. Try for about 5-10 minutes to write an inductive proof and see where you get stuck. For this part, just write "I spent about X minutes trying" (as long as you report an X at least 5, you will get this point). [1 point]

(b) Give an example of numbers a, b such that $a < b$ but $4a + 2 \not< 4b$. [1 point]
Hint: You might want to choose non-integer numbers.

Now imagine that for some number m, $f(m) = a$ and $g(m) = b$. If there were such a number m, it would be very hard to prove the inductive step of this proof! It would be impossible – from your calculation in part b, $P(m+1)$ would be false then! This observation is an indication that our $P()$ isn't strong enough. Our $P()$ **is** true for the particular f, g in this problem! But unless our $P()$ "rules out" the existence of that value m we won't be able to write the proof. Take a moment to ponder this observation – it's the main point of this problem, and somewhat mind-bending. (Don't write anything for this paragraph.)

(c) Let's try to really prove the claim now. Define the predicate $Q(n)$ to be "$g(n) \leq f(n) - 1$". Explain why showing $Q(n)$ for all $n \geq 1$ will also show $P(n)$ for all $n \geq 1$ (1 sentence) [2 points]

(d) Prove $Q(n)$ holds for all integers n, with $n \geq 1$ by induction. [20 points]

(e) This proof technique is called "strengthened induction" (not to be confused with strong vs. weak induction). We wanted to show $P()$, but $P()$ wasn't suitable for induction, the I.H. was not enough information to easily prove our target in the I.S.. We defined a strengthened claim, $Q()$. (we call $Q()$ "stronger" because once you know $Q()$ is true you also know $P()$ must be true). Choosing what to add is a tradeoff – the more information in $Q()$ that you add, the more information you have in your inductive hypothesis to assume. But also the more you have to show in your inductive step! You do not have to write anything for this part. [0 point]

4. Find. The. Bug. [7 points]

Recall the definition of **Trees** we used in class:
Basis Step: $\bullet$ is a **Tree**.
Recursive Step: If L and R are **Trees**, then $\texttt{Tree}(\bullet, L, R)$ is a **Tree**.

And recall the following definition of height:

$\text{height}(\bullet) = 0$
$\text{height}(\texttt{Tree}(\bullet, L, R)) = 1 + \max\{\text{height}(L), \text{height}(R)\}$

And the definition of leaves:
$\text{leaves}(\bullet) = 1$
$\text{leaves}(\texttt{Tree}(\bullet, L, R)) = \text{leaves}(L) + \text{leaves}(R)$ Your friend wants to show $\forall$**Trees** T, $\text{leaves}(T) = 2^{\text{height}(T)}$.

Your friend decided to use strong induction (structural induction would have been a better choice). Here is their proof.

① Define $P(n)$ to be: "all **Trees** of height n have 2^n leaves". We show $P(n)$ for all $n \geq 0$ by induction on n.

② Base Case ($n = 0$)
Ⓐ Consider an arbitrary tree of height 0, there is only one such tree $\bullet$.
Ⓑ leaves$(\bullet) = 1 = 2^0 = 2^{\text{height}(\bullet)}$.

③ Inductive Hypothesis: Suppose $P(0) \wedge \cdots \wedge P(k)$ for an arbitrary $k \geq 0$.

④ Inductive Step:
Ⓐ Let T_1 and T_2 be arbitrary **Trees** of height k. By IH applied to each, we have: leaves$(T_1) = 2^k$, leaves$(T_2) = 2^k$.
Ⓑ Define $T = \texttt{Tree}(\bullet, T_1, T_2)$.
Ⓒ To make a **Tree** of height $k + 1$, we must use the recursive rule.
Ⓓ Therefore, since T_1 and T_2 were arbitrary, T is an arbitrary **Tree** of its height.
Ⓔ height$(T) = 1 + \max\{k, k\} = k + 1$ and leaves$(T) =$ leaves(T_1) + leaves$(T_2) = 2^k + 2^k = 2^{k+1}$
Ⓕ Since T is arbitrary **Tree** of height $k + 1$, and it has $2^{\text{height}(T)} = 2^{k+1}$ leaves, we have $P(k + 1)$

⑤ Therefore $P(n)$ is true for all $n \geq 0$ by the principle of induction.

⑥ Observe that every **Tree** has height at least 0, so we have for all **Trees** T, leaves$(T) = 2^{\text{height}(T)}$.

(a) The claim is false. Identify a counter-example. You should (1) draw (or otherwise describe) the example, (2) show that it is a **Tree**(e.g. by showing the rules to build it), (3) state its height and number of leaves. If you're using LaTeX, the command for the dot is `\bullet`. [3 points]

(b) Identify the biggest flaw in the proof. We have labeled the sentences to help you describe where it goes wrong. [4 points]

5. Recursion – See: Recursion [12 points]

For each of the following languages, give a recursive definition of the language.

Your basis step must **explicitly** enumerate a finite number of initial elements. Explicitly means, for example, a regular expression is not permitted.

We may deduct for constructions that are needlessly complicated (e.g. more base cases than necessary, or significantly more recursive steps than necessary).

Briefly (1-2 sentences) justify that your description defines the same language. Do not give us a full proof; you do not have to justify why your description is the shortest possible.

(a) Binary strings that start with 0 and have even length (i.e. an even number of characters).

(b) Binary strings with an odd number of 1s.

6. Constructing Regular Expressions (Online) [20 points]

For each of the following languages, construct a regular expression that matches exactly the given set of strings.

You will submit (and check!) your answers online at https://grin.cs.washington.edu/. Think carefully before entering a submission; you only have 10 guesses. Because these are auto-graded, we will not award partial credit.

(a) Strings over $\{a, b, c\}$ where every a is immediately followed by a b or c.

(b) Binary strings where every instance of 11 is followed by 00.

(c) Strings over $\{a, b\}$ with an even number of b's

(d) The set of all binary strings that begin with a 1 and which have length ℓ such that $\ell\%4 = 3$

7. Context Is Everything. Except for Context-Free Grammars (Online) [15 points]

For each of the following languages, construct a context-free grammar that generates exactly the given language.

You will submit (and check!) your answers online at https://grin.cs.washington.edu/. Think carefully before entering a submission; you only have 10 guesses. Because these are auto-graded, we will not award partial credit.

(a) $\{a^x b^y a^{3x+y} : x, y \geq 0\}$

(b) The set of all strings over $\{a, b\}$ that contain at least one b and and most three a's.

(c) Binary strings with an odd number of 0's

8. Feedback [1 point]

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- Have you filled out the final exam information form?
- How many hours did you spend working on this assignment?
- Which problem did you spend the most time on?
- Any other feedback for us?